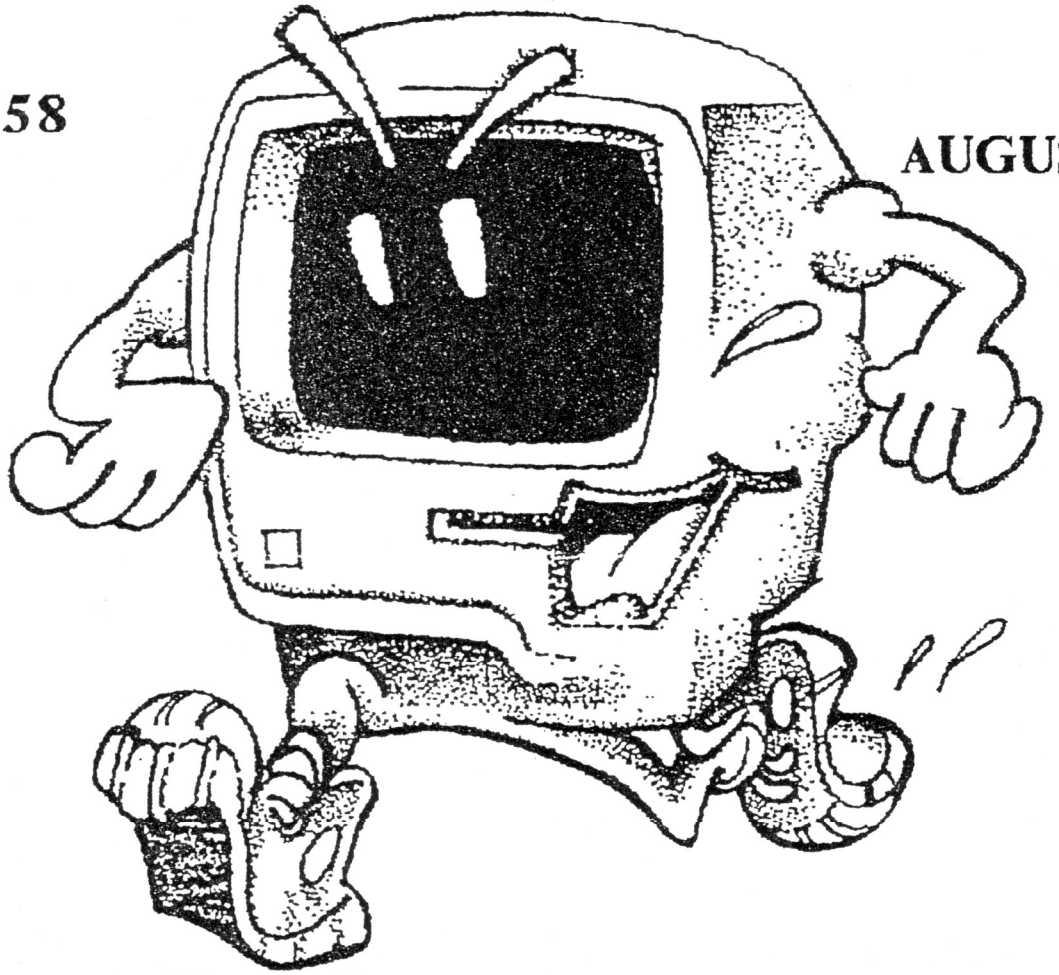
VZ LINK magazine

ISSUE 58

AUGUST 1991



A User Group is the single most important source of information for computer enthusiasts as well as novices. If you have a problem it's likely that a member of the Group will have already encountered it.

NEW ZEALAND

VZ 300/200
COMPUTER USERS CLUB

ODDS AND ENDS

Well the warmer weather is finally starting to arrive, now is the time to send me all those programs and items that you have written over the winter months.
I hope you enjoyed the July newsletter. I am always pleased to receive feed back from the members and I even appreciate the comments re spelling errors etc.

REMEMBER

If you want to contact me about any problems the phone number is now: Ph. 520-6743
If it is really urgent you can contact me at work, the number is now: Ph. 270-5389

CLUB MEETING

The remaining club meetings for the year will be now be held on :-

```
>>>>>>>>>>>>>>>>>>>>  *    SATURDAY 28 SEPTEMBER 1991    *  <<<<<<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>>>       SATURDAY 30 NOVEMBER 1991       <<<<<<<<<<<<<<<<<<<<<<<<
>>>>>>>>>>>>>>>>>>>>          Starting at 10.00 am.         <<<<<<<<<<<<<<<<<<<<<<<<
```

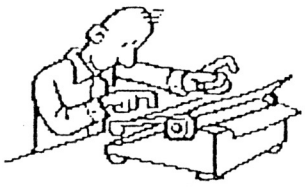
Please note the change of date for the SEPTEMBER meeting. The date shown in the last newsletter was wrong. (Next time I will turn the calender page, Brenda).

These dates are now confirmed, please mark them on you calendar. It would be really good to get a 50% turnout at least once this year.
The venue remains the same for the September meeting, hopefully we will have a new venue for November.

FOR SALE

Steelo Lemon soap pads, box of 10 for $1.00
Cassette Leads approx 1 metre long for $7.00 each (suppled by Ken Clarke)
Disks for $1.00 each /Cassettes for $2.00 each.
All proceeds to club funds.

@@@



Potpourri

by

Ken Clarke.

A potpourri of ideas, software modifications, help-line, your ideas, suggestions, gripes, where to find it, cheats, pokes and infinite lives, adventure gaming help-line, hardware help in construction, or in fact anything that other VZ'ers could be interested in or be able to find help for you.....

'SOUND EFFECTS' REVISITED. Ever notice a peculiarity when running the 'Sound Effects' program from VZ-LINK #44 [June '90]? You didn't? Give the DIRECT COMMAND - RUN "RAY GUN" , and the disk whirrs, and then follows a rather bell-like quality to the notes that are heard. However, BREAK the program and then RUN , and out rasps a slower, raucus, buzzing version!!! Go ahead and try it! Why the difference? David Lambourn explains it this way. If the VZ doesn't get its sound routine in this program continually INTERRUPTED, it will make good clean notes. But, when it has to be INTERRUPTED to attend to up-dating the screen display, then the VZ literally "BUZZES" back and forth between the two jobs of sound making and screen up-dating. With the above program, experiment. Try adding a PRINT statement. A CLS will also have the same effect. Re-SAVE and RUN "RAY GUN" [second version] again. You will find that anything to do with up-dating the screen display, starts off the INTERRUPTS again. To overcome this "BUZZING BEASTIE" you don't have to resort to avoiding using the screen and always using RUN "program name". David's formula is this. Do whatever you want to the screen but before handing control over to the SOUND ROUTINE - DISABLE THE INTERRUPT.

To DISABLE THE INTERRUPT:-

```
(100 POKE 30716,243: POKE 30717,201
(110 POKE 30862,252: POKE 30863,119: X=USR(X)
```

For convenience, this very short piece of Machine Code has been stored at the top of the Display Area [assuming MODE (1) is not required, which makes this area available. Otherwise, another location would be necessary.]

********** GET THE GENERAL BUZZ OF IT? **********

CLUB FUNDS

There are still a lot of out standing subscriptions due. If yours is one of them please forward it as soon as possible. The club relies on these to continue the newsletters. If you are not sure when and if yours is due contact me and I will tell you.

See you all in september, I hope. Regards peter.

P.O BOX 1972 C.P.O AUCKLAND N.Z

AUCKLAND VZ 300/200 COMPUTER USERS CLUB (1986)
NEW ZEALAND VZ 300/200 COMPUTER USERS CLUB (1989)

ALL MATERIAL IN THIS NEWSLETTER IS SUBJECT TO COPYRIGHT.
CONTRIBUTED PROGRAMS AND ARTICLES CAN BE REPRODUCED ON THE UNDERSTANDING THAT IT IS FOR THE PRIVATE USE OF SUBSCRIBERS ONLY.

COPYRIGHT IS ALWAYS RETAINED BY THE AUTHOR.
THIS CLUB WILL NOT BE HELD RESPONSIBLE FOR ANY BREACHING OF COPYRIGHT CAUSED BY THE PUBLICATION OR USE OF ANY MATERIAL SENT IN UNDER A FALSE NAME.

YOU ARE WELCOME TO PHONE ME ON.....PHONE 520-6743

LIVENUP

PART THREE

by Bob Kitch.

```
10 '*************************
20 '***     SET-UP        ***
30 '*** PROGRAM I OF III  ***
40 '*** TO LOAD 24 HI-RES ***
50 '*** SCREENS INTO 64K  ***
60 '*** MEM. EXP.         ***
70 '***  BOB KITCH  6/88  ***
80 '*************************
98 '
99 '***PUT UP INTRO MESSAGE.
100 GOSUB 620
110 PRINT"THIS IS THE FIRST OF A SERIES OFSUBPROGRAMS DESIGNED";
120 PRINT" TO ALLOW 24HI-RES SCREENS TO BE LOADED AND STORED ";
130 PRINT"IN A 64K MEMORY EXP. THE 2K SCREENS ARE LOADED ";
140 PRINT"VIA DOS.  THE SCREENS ARE STORED IN MEM.  BANKS 1 TO";
150 PRINT" 3 OF THE MEMORY      EXPANSION."
160 GOSUB 600
200 PRINT" <SET-UP> LOWERS TOM TO BFFFH - THE TOP OF BANK 0."
210 PRINT"THE DOS VECTOR IS ALSO LOWERED.":PRINT
220 PRINT" <LOAD-UP> SETS THE BLOCK MOVE  SECTION OF M/L IN ";
230 PRINT"PLACE. ALSO   THE 24 HI-RES SCREENS ARE CALLEDFROM ";
240 PRINT"DATA STATEMENTS. USUALLY   THESE EXIST ON A ";
250 PRINT"<PICTURE DISK> WHILST THE SUBPROGRAMS ARE ON A ";
260 PRINT"<PROGRAM DISK>"
270 GOSUB 600
300 PRINT" <THROW-UP> AUTOMATICALLY PAGES THROUGH THE 24 ";
310 PRINT"SCREENS. THIS IS SIMILAR TO <PICTURE SHOW>."
320 PRINT"THE SPEED AT WHICH THE PAGING   OCCURS CAN BE VARIED."
330 PRINT"HI-SPEED ANIMATION IS QUITE     POSSIBLE."
340 PRINT:PRINT"ALTERNATIVELY, FANCY SCREEN     MOVES CAN BE ";
350 PRINT"SELECTED."
360 GOSUB 600
400 PRINT"THE SUBPROGRAMS CHECK FOR THE   64K MEM. EXP. AND DOS"
410 PRINT:PRINT"TWO A/L PROGRAMS ARE USED."
420 PRINT:PRINT"<START-UP> LOWERS THE DOS       VECTOR."
430 PRINT:PRINT"<MOVE-UP> CONTAINS THE FANCY    SCREEN MOVES."
460 GOSUB 600
498 '
499 '***RUN LOAD-UP PROGRAM.
500 PRINT"LOADING START-UP & LOAD-UP"
510 BRUN"STARTUP"
598 '
599 '***SUBROUTINE FOR NEW SCREEN.
600 PRINT@480,"PRESS ANY KEY TO CONTINUE";
610 A$=INKEY$:A$=INKEY$:IF A$="" THEN GOTO 610
620 CLS:PRINT@13,"LIVENUP":PRINT:PRINT:RETURN
999 '
1000 '***CALLING SEQUENCE THROUGH PROGRAMS.
1010 'RUN T:SETUP   CALLS...
1020 '  BRUN B:STARTUP    CALLS...
1030 '    RUN T:LOADUP        CALLS...
1040 '      BLOAD B:MOVEUP
1050 '        RUN T:THROWUP
1060 '
1070 'A/L SOURCE CODE
1080 'SOURCE  W:START   OBJECT  B:STARTUP   ORIGIN  7200H
1090 'SOURCE  W:MOVE    OBJECT  B:MOVEUP    ORIGIN  0BC00H
1099 '
1100 '***LOAD MAP FOR LIVEN-UP.
1110 '0C000H-0FFFFH  - 3 BY 16K BANKS OF PICTURE BUFFERS.
```

```
1120 '0BEC8H-0BFFFH  - 310 BYTE DOS VECTOR AND BUFFER.
1130 '0BEABH-0BEB8H  - 14 BYTE G.P. BLOCK MOVE ROUTINE.
1140 '0BC00H-0BE0FH  - 528 BYTE MOVEUP ROUTINES.
1150 '0B200H-0BA00H  - 2K BYTE SCREEN BUFFER.
1160 '0AFFFH         - TOM FOR BASIC PROGRAMS.
1170 '0AFCDH-0AFFFH  - STRING SPACE.
1180 '0AF98H-0AFCCH  - STACK.
1190 ' 849DH-0AF97H  - FREE SPACE.
1200 ' 849DH- 849DH  - DIM VAR. TABLE (NONE).
1210 ' 8481H- 849CH  - VARIABLE TABLE.
1220 ' 7AE9H- 8480H  - PROGRAM STATEMENT TABLE.
1230 ' 7AE9H         - SOB FOR BASIC PROGRAMS.
1240 ' 7200H- 7233H  - STARTUP ROUTINE IN HI-RES SCREEN AREA.
1250 'THE BASIC PROGRAM AREA IS DESCRIBED FOR <THROWUP>.
9998 '
9999 '***UPDATE DISK FILE.
10000 ERA"SETUP":SAVE"SETUP"
20000 END
```

```
10 '*************************
20 '***     LOAD-UP       ***
30 '*** PROGRAM II OF III ***
40 '*** TO LOAD 24 HI-RES ***
50 '*** SCREENS INTO 64K  ***
60 '*** MEM. EXP.         ***
70 '***  BOB KITCH  6/88  ***
80 '*************************
898 '
899 '***START SUBPROGRAM.
900 GOTO 1100
998 '
999 '***LOAD 2K SCREEN FILE.
1000 BLOAD"ABCDEFGH":            :'***POKE ADDR. 31825 TO 31832
1010 GOTO 2220
1020 '***DO NOT ALTER ANY CODE ABOVE THIS LINE.
1030 '***OTHERWISE CHANGE AD% IN LINE #2100
1098 '
1099 '***CHECK THAT 64K MEM. EXP. OPERATIONAL.
1100 GOSUB 3020
1110 FL%=0                       :'***FLAG FOR TEST CONDITIONS.
1120 FOR I%=1 TO 3               :'***BANKS 1 TO 3.
1130    OUT 127,I%:SOUND 9,2     :'***I/O PORT 7FH.
1140    POKE -1000,(100-I%)      :'***TEST ADDR. 0FC18H.
1150 NEXT I%
1160 FOR I%=3 TO 1 STEP -1       :'***BANKS 3 TO 1.
1170    OUT 127,I%:SOUND 9,2
1180    IF PEEK(-1000)<>(100-I%) THEN FL%=-1
1190 NEXT I%
1200 IF FL%<>0 THEN PRINT@64,"64K RAM NOT FITTED.":STOP
1210 IF FL%=0 THEN PRINT@64,"64K RAM OK"
1298 '
1299 '***CHECK THAT DOS OPERATIONAL BY CHECKING FIRST 4 ID BYTES
1300 IF (PEEK(16384))<>170 THEN FL%=-1       :'***<0AAH>@4000H
1310 IF (PEEK(16385))<>85 THEN FL%=-1        :'***<55H>@4001H
1320 IF (PEEK(16386))<>231 THEN FL%=-1       :'***<0E7H>@4002H
1330 IF (PEEK(16387))<>24 THEN FL%=-1        :'***<18H>@4003H
1340 IF FL%<>0 THEN PRINT@128,"DOS NOT FITTED.":STOP
1350 IF FL%=0 THEN PRINT@128,"DOS OK"
1398 '
```

```
1399 '***CHECK TOM IS 0BEC8H.
1400 IF (PEEK(30897))<>200 THEN FL%=-1
1410 IF (PEEK(30898))<>190 THEN FL%=-1
1420 IF FL%<>0 THEN PRINT@192,"TOM NOT SET CORRECTLY":STOP
1430 IF FL%=0 THEN PRINT@192,"TOM OK"
1496 '
1497 '***LOWER TOM TO 0AFFFH FOR BUFFER, M/L AND JUMP POINTERS.
1498 '***0B200H-0BBFFH IS A SCREEN BUFFER.
1499 '***0BC00H-0BE0FH IS M/L MOVE ROUTINES LOADED FROM "MOVEUP"
1500 POKE 30897,255:POKE 30898,175:CLEAR 50
1510 PRINT:PRINT"LOADING MOVE-UP ROUTINE."
1520 BLOAD"MOVEUP":                    :'***LOAD SCREEN MOVE SUBS.
1598 '
1599 '***POKE IN BLOCK MOVE FROM 0BEABH. (0BE10H-0BEAAH SPARE.)"
1600 DATA 243          :' DI
1610 DATA 033,000,112  :' LD HL,7000H     (SOURCE)
1620 DATA 017,000,192  :' LD DE,C000H     (DESTINATION)
1630 DATA 001,000,008  :' LD BC,0800H     (SIZE)
1640 DATA 237,176      :' LDIR
1650 DATA 251          :' EI
1660 DATA 201          :' RETURN
1670 FOR AD%=-16725 TO -16712     :'***FROM 0BEABH TO 0BEB8H.
1680    READ DT%:POKE AD%,DT%
1690 NEXT AD%
1698 '
1699 '***SET USR VECTOR TO 0BEABH.
1700 POKE 30862,171:POKE 30863,190
1798 '
1799 '***SET ADDR. FOR MSB OF DESTINATION IN M/L FOR PAGING.
1800 MD%=-16719                    :'***MSB OF DEST. PG. 0BEB1H.
1898 '
1899 '***PAUSE AND PUT UP MESSAGE.
1900 GOSUB 3020:PRINT"A COUPLE OF OPTIONS EXIST HERE."
1910 PRINT:PRINT" <1> CONTINUE WITH 2K BY 24 PIC FILES."
1920 PRINT:PRINT" <2> DO 16K BY 3 MEM BANK LOAD.":PRINT
1930 PRINT"ENTER 1 OR 2";:INPUT A$:POKE 30791,ASC(A$)
1940 IF A$="1" THEN GOTO 1960
1950 IF A$="2" THEN GOTO 2400 ELSE GOTO 1900
1960 GOSUB 3020
1970 PRINT"    ***INSERT PICTURE DISK***"
1980 SOUND8,5;0,3;8,5;0,3;8,5
1990 GOSUB 3000
1998 '
1999 '***DATA FOR 24 SCREEN FILES.
2000 '***MEMORY BLOCK 1.
2010 DATA 1.1.EFGH,1.3.EFGH,1.2.EFGH,1.4.EFGH
2020 DATA 1.5.EFGH,1.6.EFGH,1.7.EFGH,1.8.EFGH
2030 '***MEMORY BLOCK 2.
2040 DATA 2.1.EFGH,2.3.EFGH,2.2.EFGH,2.4.EFGH
2050 DATA 2.5.EFGH,2.6.EFGH,2.7.EFGH,2.8.EFGH
2060 '***MEMORY BLOCK 3.
2070 DATA 3.1.EFGH,3.3.EFGH,3.2.EFGH,3.4.EFGH
2080 DATA 3.5.EFGH,3.6.EFGH,3.7.EFGH,3.8.EFGH
2098 '
2099 '***MAIN READ-IN ROUTINE.
2100 AD%=31825:MODE(1)
2110 FOR I%=1 TO 3                 :'***MEM. BLOCKS 1 TO 3
2120    OUT 127,I%                 :'***SET MEMORY BANK.
2130    FOR J%=192 TO 255 STEP 8   :'***MSB 0C0H,0F8H,8H(8 SCRNS)
2140       POKE MD%,J%             :'***SET MSB OF PAGE IN M/L.
2150       F$=""                   :'***ASSISTS STRING HANDLING.
2160       READ F$:L%=LEN(F$):IF L%>8 THEN F$=LEFT$(F$,8):L%=8
2170       IF L%<>8 THEN F$=LEFT$("        ",(8-L%))+F$
2180       FOR K%=1 TO 8           :'***POKE IN FILENAME.
2190          POKE AD%-1+K%,ASC(MID$(F$,K%,1))
2200       NEXT K%
```

```
2210      GOTO 1000                :'***LOAD FILE FROM DISK.
2220    K%=USR(0)                  :'***MOVE SCREEN TO HI-MEM.
2230    NEXT J%
2240 NEXT I%
2298 '
2299 '***DISPLAY END MESSAGE.
2300 MODE(0)                       :'***RETURN TO TEXT MODE.
2310 GOSUB 3020
2320 PRINT@64,"ALL SCREENS HAVE BEEN SAVED TO  THE MEMORY ";
2330 PRINT"EXPANSION."
2340 PRINT"REINSERT THE <PROGRAM DISK> TO  CONTINUE DISPLAY"
2350 SOUND 8,5;0,3;8,5;0,3;8,5
2360 PRINT@320,"IF <MEMORY BANK SAVE> WANTED-"
2370 PRINT"THEN CONTINUE"
2380 GOSUB 3000
2398 '
2399 '***RUN THROW-UP DISPLAY PROGRAM.
2400 PRINT:PRINT"LOADING THROW-UP"
2410 RUN"THROWUP"
2997 STOP
2998 '
2999 '***SUROUTINE FOR NEW SCREEN.
3000 PRINT@480,"[illegible]";
3010 A$=INKEY$:A$=INKEY$:IF A$="" THEN GOTO 3010
3020 CLS:PRINT@13,"[illegible]":PRINT:PRINT:RETURN
9998 '
9999 '***UPDATE DISK FILE.
10000 ERA"LOADUP":SAVE"LOADUP"
20000 END
```

LOADING OF SYSTEM 80/TRS 80 FILES TO THE VZ

By Pete Patterson

Unfortunately the article in the July VZ LINK on this very useful utility was a bit screwed up. No fault of Ken though,as the instructions I received with the XX-80 utility are quite misleading and it took some considerable time to get the utility to work. During this time, Peter and Ken got a fair bit of garbled info from me about the loader.

There are a couple of versions about for the VZ. A later version,automatically seeks top of memory of any configuration of VZ's. Unfortunately the one I have, places itself into the top of memory of a VZ200 with a 16K memory expansion only.(No disk drive or controller fitted.) Please don't ask how long it took for me to find this out.The instructions certainly don't tell you.

To load the XX-80,type CLEAR200:CLOAD and press RETURN.

Once the utility is loaded,type NEW and press RETURN. Then enter PRINTUSR(0) and press RETURN. The VZ200 will show on the bottom left of your screen LOADING XX80-FILE. You now load the system 80 file using the Data Recorder. When loading is completed the normal READY appears. If you have a Data Recorder with a volume control you will most likely have to adjust the level to get a good load. Once the program is loaded LIST to see how well it loaded and if OK,CSAVE onto tape. You can then later reload for saving onto disk and for modifying to run on the VZ.

The loader works well and once you get the correct setting of your data recorder volume control there is very little corruption.

There must be plenty of TRS-80/SYSTEM80 tapes around that are suitable for conversion for running on the VZ.

INTERRUPTS

Reprinted from VZ USER magazine with permission of Mark Harwood.

This month, we are studying the operation of the interrupt in the VZ, and how to write an interrupt routine.

In the VZ, the VDG (Video Display Generator) generates an interrupt signal at the end of every screen scan completion.
This occurs approximately every one fiftieth of a second.
This signal is fed to the INT pin of the Z80A, and, providing interrupts are enabled, the Z80 will call the appropriate action according to which interrupt mode the CPU is in at that moment.

The Z80 can be placed in any of three interrupt modes - MODE 0,1 or 2.
These are selected using the IM0, IM1 and IM2 instructions respectively.
All interrupts can be disabled using the DI instruction, and hence the INT interrupt is known as a maskable interrupt.

Interrupt mode 1 is the only mode that can be used on the basic VZ. This is because both modes 0 and 2 require the device which generated the interrupt to supply an instruction of one sort or another. Since no other hardware interrupts are used in the VZ, these are not used.

Mode 1 is a vectored interrupt. As soon as the INT signal occurs, the Z80 stores the contents of the program counter onto stack, and then carries out a restart to address 0038H. This is of course in ROM. This location contains a JP 2EB8H instruction. The routine found at 2EB8H is listed below:

```
2EB8:F5          PUSH AF                ; save registers
2EB9:C5          PUSH BC
2EBA:D5          PUSH DE
2EBB:E5          PUSH HL
2EBC:CD7D78      CALL 787DH             ; call users interrupt
                     .
                     .
                     .
                     .
                     .
2ED5:E1          POP HL                 ; get registers back
2ED6:D1          POP DE
2ED7:C1          POP BC
2ED8:F1          POP AF
2ED9:FB          EI                     ; enable interrupts
2EDA:ED4D        RETI                   ; return from inter.
```

Firstly, the registers are saved. A CALL is then made to 787DH (30845) which is located in the ' Communications ' region of User RAM, and is listed in the Technical Reference Manual as the interrupt exit.
Location 30845 usually contains C9H (201), which represents a RET instruction.
The CALLs between 2EBDH and 2ED4H are calls to routines which update the screen, cursor and scan the keyboard. The registers are then restored, and the routine returns to wherever the counter was using the RETI instruction.

Thus, to implement our own interrupt-driven routine, the RET at 30845 must be changed to a JP instruction (C3H) to the start of the routine.

The two routines we've included here are an example of how an interrupt routine can be implemented. Our routines give your VZ a 'Clear screen key'.
The first version is intended for use with the function keys which appeared in issue 45, July 1990 and the second uses SHIFT-C as the 'cls' key.
As soon as the clear key is pressed, the screen is cleared, independent of what the CPU is doing at the time (except tape operations-when the interrupt is disabled).
The same structure can be adopted with most other interrupt routines.

(Continued over page)

We assembled both programmes using the Editor/Assembler. After entering the code, save the source, just in case you,ve made an error.
Set the origin to the appropriate value for your particular memory size, dependant on machine and RAM packs installed.
Assemble the object, and save it. The programme is located at the top of memory, and sets the TOM pointer when it auto-executes on loading. The CPU has no knowledge of its existence.

To use the 'cls' key function, load the object, which should only take a few seconds. Type NEW as soon as it's ready - this stabilises the other pointers.
To activate the routine, enter POKE 30845,195 (This replaces the RET with a JP).
Similarly, to disable it, POKE 30845,201.
You might like to make up a whole set of extra function keys, using those SHIFT-X combinations currently unused on the VZ.

Here's a listing of the second version, using SHIFT-C:

```
            ORIGIN: APPROX. 55 BYTES BELOW TOM
        For Min. VZ200  : 036803
         "  VZ200 + 16K : 053190
         "  Min. VZ300  : 047043
         "  VZ300 + 16K : 063247

STRT  PUSH HL              ; save registers
      PUSH AF
      LD HL,INT            ; set int. pointer
      LD (30846),HL        ; to start of actual routine
      LD HL,STRT-1         ; set TOM ptr. to before start
      LD (30897),HL
      POP AF               ; restore registers
      POP HL
      JP 1A19H             ; jump to BASIC
INT   PUSH AF              ; save registers
      PUSH BC
      PUSH DE
      PUSH HL
      LD A,(68FBH)         ; get row address of SHIFT-C
      OR 243               ; SHIFT & C pressed ?
      CP 243
      CALL Z,CLS           ; clear screen if so
      POP HL               ; restore registers
      POP DE
      POP BC
      POP AF
      RET                  ; return to main interrupts
CLS   LD BC,200H           ; no. of bytes to fill
      LD HL,7000H          ; start screen memory
LOOP  LD A,20H             ; code for space
      LD (HL),A            ; store to screen address
      INC HL               ; bump pointer
      DEC BC               ; drop counter
      LD A,C               ; counter=0 ?
      OR B
      JR NZ,LOOP           ; loop if not
      RET                  ; return to main routine
```

For the first version, using the special function key in issue 45, July 1990 simply replace LD A,(68FEH) (for function key's row address), and replace both the OR and CP which follow with CP 251.

PRINT STYLES

by Lou Ellmers

PROGRAM TO SET PRINTER STYLE

```
1 CLS
2 PRINT"        SET PRINTER STYLE";:INPUTX:CLS
10  PRINT"SELECT ITALIC CHRS.          ( 1)"
20  PRINT"CANCEL ITALIC                ( 2)"
30  PRINT"SELECT EMPHASIZED PRINT      ( 3)"
40  PRINT"CANCEL EMPHASIZED PRINT      ( 4)"
50  PRINT"SELECT CONDENSED PRINT       ( 5)"
60  PRINT"CANCEL CONDENSED PRINT       ( 6)"
70  PRINT"SELECT DBLE/WIDTH ONE LINE ( 7)"
80  PRINT"CANCEL DBLE/WIDTH ONE LINE ( 8)"
90  PRINT"SELECT DOUBLE STRIKE         ( 9)"
100 PRINT"CANCEL DOUBLE STRIKE         (10)"
105 PRINT"ENTER NUMBER";:INPUTA$
110 PRINT"SELECT SUPER SCRIPT          (11)"
120 PRINT"SELECT SUB   SCRIPT          (12)"
130 PRINT"CANCEL SUPER OR SUB SCRIPT (13)"
140 PRINT"SET N/216 LINE SPACING       (14)"
150 PRINT"SELECT MASTER STYLE COMMAND(15)"
160 PRINT"TURN   UNDERLINING ON        (16)"
170 PRINT"CANCEL UNDERLINING           (17)"
180 PRINT"SELECT DOUBLE WIDTH          (18)"
190 PRINT"CANCEL DOUBLE WIDTH          (19)"
298 PRINT"ENTER NUMBER";
299 INPUTB$
300 REM SELECT NUMBERS
301 IFA$= "1" GOSUB 501
302 IFA$= "2" GOSUB 502
303 IFA$= "3" GOSUB 503
304 IFA$= "4" GOSUB 504
305 IFA$= "5" GOSUB 505
306 IFA$= "6" GOSUB 506
307 IFA$= "7" GOSUB 507
308 IFA$= "8" GOSUB 508
309 IFA$= "9" GOSUB 509
310 IFA$="10" GOSUB 510
311 IFB$="11" GOSUB 511
312 IFB$="12" GOSUB 512
313 IFB$="13" GOSUB 513
314 IFB$="14" GOSUB 514
315 IFB$="15" GOSUB 515
316 IFB$="16" GOSUB 516
317 IFB$="17" GOSUB 517
318 IFB$="18" GOSUB 518
319 IFB$="19" GOSUB 519
320 IFC$="20" GOSUB 520
321 IFC$="21" GOSUB 521
322 IFC$="22" GOSUB 522
323 IFC$="23" GOSUB 523
324 IFC$="24" GOSUB 524
500 GOTO1000
501 LPRINTCHR$(27);CHR$(52);:  RETURN :          '( 1)
502 LPRINTCHR$(27);CHR$(53);:  RETURN :          '( 2)
503 LPRINTCHR$(27);CHR$(69);:  RETURN            '( 3)
504 LPRINTCHR$(27);CHR$(70);:  RETURN :          '( 4)
505 LPRINTCHR$(27);CHR$(15);:  RETURN :          '( 5)
```

```
506 LPRINTCHR$(18);:              RETURN :            '( 6)
507 LPRINTCHR$(27);CHR$(14);:     RETURN :            '( 7)
508 LPRINTCHR$(20);:              RETURN :            '( 8)
509 LPRINTCHR$(27);CHR$(71);:     RETURN :            '( 9)
510 LPRINTCHR$(27);CHR$(72);:     RETURN :            '(10)
511 LPRINTCHR$(27);CHR$(83);CHR$(48);:RETURN :        '(11)
512 LPRINTCHR$(27);CHR$(45);CHR$(49);:RETURN :        '(12)
513 LPRINTCHR$(27);CHR$(84);:     RETURN :            '(13)
514 LPRINTCHR$(27);CHR$(51);CHR$(15);:RETURN :        '(14)
515 LPRINTCHR$(27);CHR$(33);CHR$(4);:RETURN :         '(15)
516 LPRINTCHR$(27);CHR$(45);CHR$(49);:RETURN :        '(16)
517 LPRINTCHR$(27);CHR$(45);CHR$(48);:RETURN :        '(17)
518 LPRINTCHR$(27);CHR$(87);CHR$(49);:RETURN :        '(18)
519 LPRINTCHR$(27);CHR$(87);CHR$(48);:RETURN :        '(19)
520 'LPRINTCHR$(27);:RETURN
521 'LPRINTCHR$(27);:RETURN
522 'LPRINTCHR$(27);:RETURN
523 'LPRINTCHR$(27);:RETURN
524 'LPRINTCHR$(27);:RETURN
1000 REM ANOTHER NUMBER
1010 PRINT"ANOTHER NUMBER Y/N?";:INPUT B$
1020 IFB$="Y" GOTO 10 ELSE 1030
1030 LPRINT"PROGRAM TO SET PRINTER STYLE"
1040 LPRINT"BY LOU ELLMERS.":LPRINT
1050 PRINT"AGAIN? Y/N?":INPUTC$:IFC$="Y"THEN10ELSEEND
```

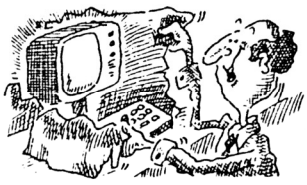
This program is designed to allow different print styles to be set for your printer. I am not sure if it will work for other printers as they may have different commands for the same things. Check your printer manual if you are not sure.

The listing is a little untidy, I tried to change line numbers and got into a awful tangle. I would have been better to start again and forget the first lot.
Anyway, here is a printout of my efforts. It took a while to sort out the traps we novices fall into. This version is about Mark 7.

A easy trap to miss is having spaces in the wrong place ie.
I had lines 301 to 309 with a gap between the " and the number like so:- " 5"
It should of been "5". No wonder it wouldn't work until it got to 10. A simple one but easy to over look.

The last three lines will need to be modified for your own use, but I am sure most of you are able to do this. Lines 520 to 524 are for extending later on.

(A example of print styles can be found on the back page..Peter)



"Make things as simple as possible.
But no simpler" Albert Einstein

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.

PROGRAM TO SET PRINTER STYLE
BY LOU ELLMERS.